

VŠB – Technická univerzita Ostrava
Fakulta strojní
Katedra automatizační techniky a řízení

Zobrazování veřejných dat s využitím platformy Arduino

Display of Public Data Using the Arduino platform

Student:

Jiří Poštulka

Vedoucí Bakalářské práce:

Ing. Pavel Smutný, Ph.D.

Ostrava 2018

Zadání bakalářské práce

Student: **Jiří Poštulka**
Studijní program: **B2341 Strojírenství**
Studijní obor: **3902R001 Aplikovaná informatika a řízení**
Téma: **Zobrazování veřejných dat s využitím platformy Arduino**
Display of Public Data Using the Arduino Platform
Jazyk vypracování: **čeština**

Zásady pro vypracování:

1. Seznamte se s vývojovou deskou Arduino a možnostmi připojení displeje a wi-fi modulu.
2. Proveďte analýzu a srovnání již realizovaných řešení úloh zobrazování dat z Facebook API.
3. Navrhněte vhodné hardwarové a softwarové prostředky pro vlastní realizaci úlohy zobrazování dat z Facebook API.
4. Zhodnoťte dosažené výsledky a navrhněte směry dalšího řešení.

Seznam doporučené odborné literatury:

CZUDEK, Adam. Využití open-source hardware pro zpracování veřejných dat [online]. 2016 [cit. 2017-09-27]. Dostupné z: <http://hdl.handle.net/10084/115119>. Diplomová práce. Vysoká škola báňská - Technická univerzita Ostrava.

FACEBOOK.COM. Facebook for Developers [online]. 2014 [cit. 2014-03-07]. Dostupné z: <http://developers.facebook.com>

MARGOLIS, Michael. Arduino cookbook. 2nd ed. Sebastopol, Calif.: O'Reilly, c2012, 699 p. ISBN 1449313876.

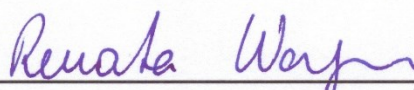
MELGAR, Enrique Ramos, Ciriaco CASTRO DÍEZ a Przemek JAWORSKI. Arduino and Kinect projects: Design, build, blow their minds. New York: Distributed to the book trade worldwide by Springer Science+Business Media, c2012, 393 p. ISBN 9781430241683.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Pavel Smutný, Ph.D.**

Datum zadání: 08.12.2017

Datum odevzdání: 21.05.2018


doc. Ing. Renata Wagnerová, Ph.D.
vedoucí katedry




doc. Ing. Ivo Hlavatý, Ph.D.
děkan fakulty

Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou bakalářskou práci včetně příloh vypracoval samostatně pod vedením vedoucího bakalářské práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě dne:

.....

Podpis studenta

Prohlašuji, že

- jsem si vědom, že na tuto moji závěrečnou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. Zákon o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (dále jen Autorský zákon), zejména § 35 (Užití díla v rámci občanských či náboženských obřadů nebo v rámci úředních akcí pořádaných orgány veřejné správy, v rámci školních představení a užití díla školního) a § 60 (Školní dílo),
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo užít tuto závěrečnou bakalářskou práci nekomerčně ke své vnitřní potřebě (§ 35 odst. 3 Autorského zákona),
- bude-li požadováno, jeden výtisk této bakalářské práce bude uložen u vedoucího práce,
- s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 Autorského zákona,
- užít toto své dílo, nebo poskytnout licenci k jejímu využití, mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše),
- beru na vědomí, že - podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů - že tato bakalářská práce bude před obhajobou zveřejněna na pracovišti vedoucího práce, a v elektronické podobě uložena a po obhajobě zveřejněna v Ústřední knihovně VŠB-TUO, a to bez ohledu na výsledek její obhajoby.

V Ostravě dne:

.....

Podpis autora práce

Jméno a příjmení autora práce: Jiří Poštulka

Adresa trvalého pobytu autora práce: Strmá 43/14, Kozmice 747 11

ANOTACE BAKALÁŘSKÉ PRÁCE

POŠTULKA, J. *Zobrazování veřejných dat s využitím platformy Arduino: bakalářská práce.* Ostrava: VŠB – Technická univerzita Ostrava, Fakulta strojní, Katedra automatizační techniky a řízení, 2018, 44 s. Vedoucí práce: Smutný, P.

Bakalářská práce se zabývá možnostmi využití open-source platformy Arduino v oblasti získávání a zobrazování veřejných dat. V teoretické části jsou popsány základní informace týkající se platformy Arduino, vývojového prostředí a možností rozšíření o další periférie. V následující kapitole jsou zveřejněny některé již realizované projekty zabírající se stejnou problematikou. Poslední částí této práce je návrh a realizace vlastního hardwaru, jenž obsahuje Arduino Yún, pět 7-segmentových displejů a posuvné registry pro komunikaci mezi Arduinem a displeji. Rovněž je řešením i software, který má za úkol načítat data z Facebook API a zobrazovat je na výše zmíněných displejích.

Klíčová slova: Arduino, Facebook API, 7-segmentový displej

ANNOTATION OF THE BACHELOR THESIS

POŠTULKA, J. *Display public data using the Arduino platform: bachelor thesis.* Ostrava: VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Control Systems and Instrumentation, 2018, 44 p. Leadership: Smutný, P.

The bachelor thesis deals with the possibilities of using open-source Arduino platform in the area of acquisition and display of public data. The theoretical part describes the basic information concerning the Arduino platform, the development environment and the possibilities of extending it with other peripherals. In the following chapters, some already implemented projects dealing with the same issue. The last part of this work is the design and implementation of own hardware, which includes Arduino Yun, five 7-segment displays and sliding registers for communication between Arduin and the display. It is also a software solution that has the task of retrieving data from the Facebook API and displaying them on the aforementioned displays.

Keywords: Arduino, Facebook API, 7-segment display

Obsah

Seznam použitých zkratk a symbolů	7
Úvod	8
1 Seznámení s Arduinem	9
1.1 Rozdělení	9
1.2 Základní typy Arduino desek:	10
1.3 Vývojové prostředí.....	12
1.4 Shieldy a moduly.....	13
2 Realizovaná řešení pro zobrazování dat z Facebook API	15
3 Návrh vlastního hardwarového zařízení.....	16
4 Analýza možností připojení do Wifi sítě VŠB-TUO	20
5 Návrh vlastního softwaru	22
5.1 Facebook API	22
5.2 PHP skript.....	23
5.3 Arduino	25
6 Kompletace a popis činnosti	29
7 Závěr	32
Seznam použité literatury	35
Seznam příloh.....	37
Příloha A – Uživatelský návod.....	38
Příloha B – PHP skript 1	42
Příloha C – PHP skript 2	42
Příloha D – Arduino kód.....	43

Seznam použitých zkratek a symbolů

IDE	Integrated development environment
API	Application programming interface
IoT	Internet of things
I/O	Input/Output
PWM	Pulse width modulation
SMD	Surface mount device
LAN	Local area network
SD	Secure digital
GPS	Global positioning system
LCD	Liquid crystal display
FB	Facebook

Úvod

Tato bakalářská práce se bude zabírat zobrazováním veřejných dat prostřednictvím open-source hardware Arduino. K získání dat z webové stránky se využívá API. Jedná se o termín softwarového inženýrství, jenž se dá pojmut jako aplikační programové rozhraní. API je chápáno jako soubor procedur, funkcí či protokolů konkrétní knihovny, které je možno využívat. Zjednodušeně řečeno se jedná o jakéhosi prostředníka, kterému zadáme, co požadujeme, on přistoupí do databáze a vrátí nám požadované informace, s kterými je možno dále pracovat.

A právě s těmito informacemi dále pracují i zařízení, která je zobrazují na různých formách displeje. S tímto typem zařízení je spojen termín internet věcí (IoT). IoT je nový trend v oblasti kontroly a komunikace předmětů denního použití mezi sebou nebo s člověkem a to zejména prostřednictvím internetu. V praxi IoT usiluje o z automatizování rutinních činností, např. většina lidí si po ránu dává kávu, v tomto případě by budík dal vědět kávovaru, že jste vstal, a ten by vám jí začal připravovat. A to je právě cílem a vizí IoT, snaha o vzájemnou komunikaci a sdílení dat jednotlivých zařízení.

Takovéto zařízení bude cílem této bakalářské práce, kdy si zařízení přečte požadovaná data z veřejného zdroje a zobrazí je na displeji. Zařízení, kterým bude tento proces řízen, je open-source platforma Arduino. Jelikož s tímto open-source hardwarem nemám žádné zkušenosti, budu se nejdříve muset s tímto zařízením seznámit.

Arduino bude pomocí API získávat veřejná data z Facebooku, přesněji půjde o počet „Like“ ze stránky Vysoké školy báňské - Technické Univerzity Ostrava. Touto realizací bude demonstrován již zmíněný princip IoT, kdy i prostřednictvím svého telefonu s připojením k internetu a Facebookovým účtem, dokážete „řídít“ reálné zařízení.

1 Seznámení s Arduinem

Arduino je open-source hardware, který se v posledních letech významně rozšířil. Obdobou open-source hardware je open-source software, u kterého je možné daný software používat, kopírovat, studovat i měnit a zdrojový kód je volně sdílen. Typickým příkladem takového to softwaru je operační systém Linux. U open-source hardwaru jde o fyzická zařízení, která můžeme studovat, upravovat a dále prodávat. A právě díky této licenci máme možnost nalézt celou řadu projektů včetně schémat zapojení a zdrojového kódu volně na internetu. (Open-source hardware, 2001)

Hlavně díky ceně se Arduino stalo mozkiem tisíce projektů po celém světě, které jsou na internetu veřejně přístupné. Původním autorem je Institut Interaktivního Designu ve městě Ivrea, kde vznikl jako nástroj pro rychlé vytváření prototypů, určených pro studenty bez hlubších znalostí programování a elektroniky. Arduino komunita se zanedlouho velmi rozrostla. Deska se začala měnit a přizpůsobovat podle nových výzev a potřeb. Z jednoduchých 8bitových desek se Arduino měnilo na produkty pro IoT (Internet of Thing) aplikace, 3D tisk nebo speciální Arduino roboty. Všechny Arduino desky i software jsou kompletně open-source. Arduino desky můžete jakkoliv využít, nebo je podle dostupných dat vyrobit. Můžete je také přizpůsobit konkrétním požadavkům a vytvořit si vlastní Arduino desku. (Arduino - Introduction, 2017)

1.1 Rozdělení

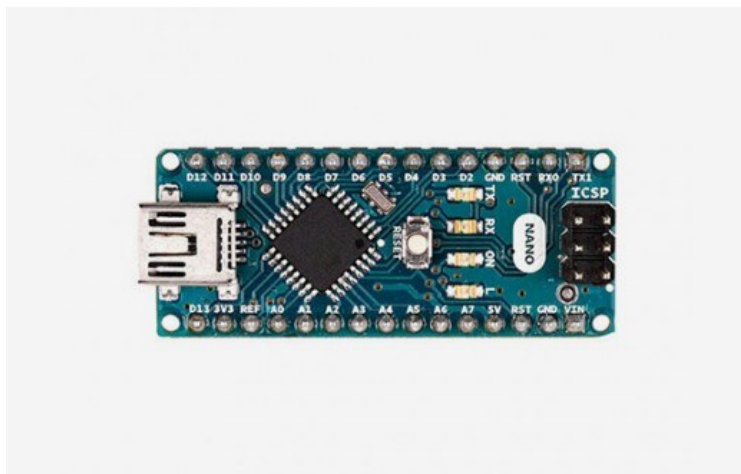
Jedná se o desku plošných spojů, na které nalezneme napájecí část, mikroprocesor, převodník pro komunikaci s počítačem a řadu pinů sloužících k připojení analogových i digitálních vstupů a výstupů. Arduino desek je několik typů, záleží čistě na potřebě projektu, který chceme realizovat. Podle funkce, kterou požadujeme, můžeme k desce připojit velkou škálu rozšiřujících desek, které jsou označovány jako Arduino shieldy. K těmto shieldům je zpravidla dodávána i knihovna pro okamžité použití. Umožňují nám připojení různých typů zařízení, např. displejů, motorů a bezdrátových modulů.

Desky jsou dostupné v celé řadě obchodů, ve kterých nalezneme originály od původního autora, ale i klony od ostatních výrobců. Originály mají certifikáty dávající jistotu, že nám vše bude fungovat přesně jak má, narozdíl od klonů. Ty mají mnohdy častější problémy se spolehlivostí, převážně se jedná o bezejmenné výrobce z Číny, tím pádem bývá cena zhruba o polovinu nižší než za originál, ovšem i u klonů lze nalézt kvalitního dodavatele.

1.2 Základní typy Arduino desek:

Arduino nano

Jedná se o nejmenší desku rodiny Arduino (18x45 mm). I přes svou velikost disponuje 32KB flash pamětí, která je využívána mikrokontrolérem Atmel ATmega328 a nalezneme zde 8 analogových I/O pinů a 22 digitálních, z nichž 6 podporují PWM.



Obrázek 1 Arduino Nano (ARDUINO NANO, 2017)

Arduino Mega

Desky řady Mega dokáží zaujmout především úctyhodnými rozměry (101x53 mm). Na této desce nalezneme mikrokontrolér Atmel ATmega2560 s 256KB flash pamětí, analogových vstupů má 16 a digitálních I/O 54 z nichž lze 15 využít jako PWM výstup.



Obrázek 2 Arduino Mega (ARDUINO MEGA 2560 REV3, 2017)

Arduino Uno

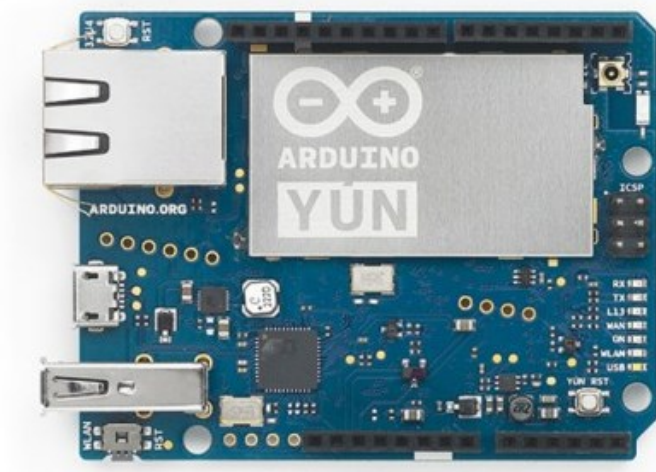
Nejspíše jedna z nejznámějších a nejvyužívanějších desek Arduino. Aktuálně jsou nabízeny tři verze této desky a hned v několika provedeních. K dispozici máme SMD i klasickou verzi, Uno s Wifi modulem a další jako například Genuino či Leonardo, které přímo vychází z desky Uno. Stejně jako nano obsahuje i tato deska mikro kontrolér Atmel ATmega328, stejně tak nabídne 32KB flash paměti. Ačkoliv je tato deska rozměrově větší oproti verzi nano, nabídne „pouze“ 6 analogových vstupních pinů, 14 digitálních I/O pinu, z nichž 6 může být použito jako PWM výstup.



Obrázek 3 Arduino Uno (ARDUINO UNO REV3, 2017)

Arduino Yún

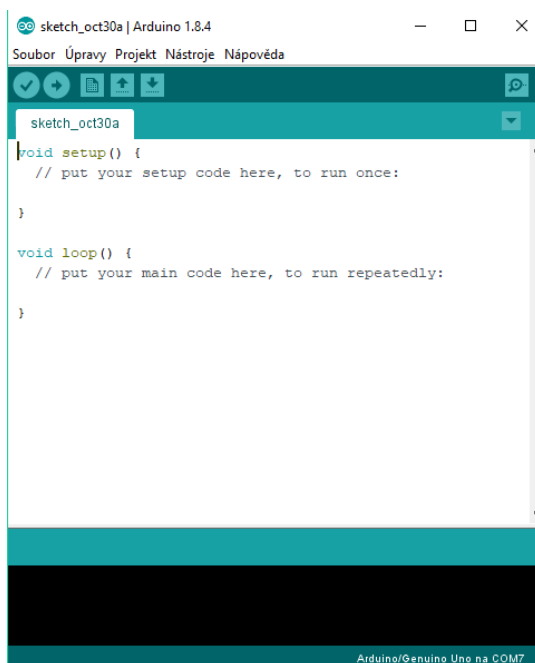
Hlavní výhodou této desky oproti ostatním řadám je možnost připojení k internetu a to pomocí kabelu či bezdrátově bez potřeby připojení shieldu. Arduino Yún v sobě kloubí jednoduchost Arduina a komplexnost Linuxu, přesněji Linuxovou distribucí OpenWRT, která je běžně využívána v řadě zařízení (např. routery). Samotné Arduino obsahuje dvě výpočetní jednotky, mikrokontroler ATmega32u4 a procesor Atheros AR9331, který se stará o Linux. Arduino Yún je tak možno nejenom připojit k internetu, ale je zde možnost vzdálené správy díky zabudovanému Wifi modulu. Poskytuje 12 analogových I/O pinu a 20 digitálních I/O pinů, z nichž může být 7 použito jako PWM výstup. (Typy Arduino desek, 2015)



Obrázek 4 Arduino Yún (ARDUINO YÚN, 2017)

1.3 Vývojové prostředí

Vývojové prostředí Arduina (IDE Arduino) je stejně tak úspěšné jako samotný hardware a to hlavně díky jeho jednoduchosti. IDE obsahuje textový editor a pár tlačítek, které už na první pohled naznačují jejich funkci. Právě kvůli velmi intuitivnímu prostředí je schopen začátečník velmi rychle psát a nahrávat své programy. Jazyk, který slouží k psaní programů, je podobný C a C++. (Slinták, 2011)



Obrázek 5 IDE Arduino

1.4 Shieldy a moduly

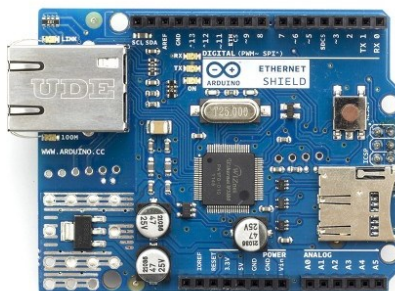
Nedílnou součástí Arduina jsou právě shieldy a moduly, které slouží k rozšíření základní desky.

Shieldy

Jedná se o nadstavbu pro Arduino desky, nejčastěji pro Uno nebo Mega. Shield jednoduše nasunete na základní desku bez jakýchkoliv dalších drátových propojení. Shieldů je celá řada, například: Ethernet shield, Wifi shield, Motor shield a mnoho dalších.

Ethernet shield

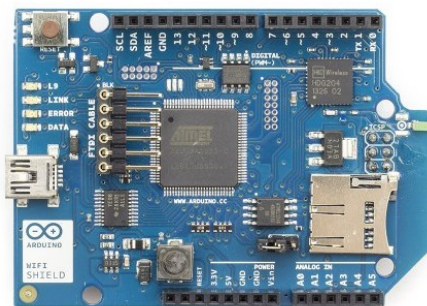
Jedná se o rozšiřující kartu, která umožní komunikaci prostřednictvím LAN sítě. Nalezneme zde klasický port pro připojení kabelu s koncovkou RJ-45, kde podpora přenosu rychlosti je 10/100Mb. Krom LAN připojení obsahuje i slot pro paměťovou kartu micro SD, na kterou je schopen ukládat různá data.



Obrázek 6 Ethernet shield (ARDUINO ETHERNET SHIELD, 2017)

Wifi Shield

Tento shield pracuje s Wifi na frekvenci 2,4 GHz ve standardu 802.11b/g. Podporované šifrování Wifi sítě je WEP a WAP2, což je pro většinu připojení dostačující. I tento shield nabídne slot pro micro SD.



Obrázek 7 Wifi shield (Arduino Wi-fi shield, 2017)

Moduly

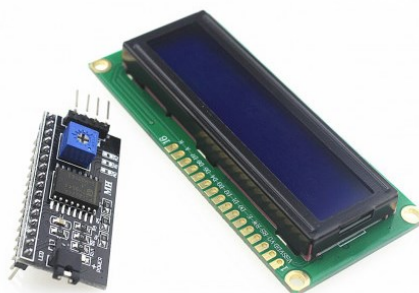
Na rozdíl od shieldů jsou moduly připojeny přes jednotlivé piny. Modulů je nepřeberné množství, a proto bych je rozdělil na určité kategorie. Moduly mohou dělit podle funkce na moduly pro komunikaci, moduly displejů, reléové moduly, moduly čidel a senzorů, moduly tlačítek a ovladačů a mnoho dalších.

Moduly pro komunikaci

V této kategorii nalezneme moduly pro připojení k internetu (Wifi a Ethernet modul), moduly pro GPS, Bluetooth moduly a celou řadu dalších. Ethernetový modul se na rozdíl od Ethernet shieldu připojuje pomocí 6 pinů, nezávisle na typu Arduino desky (u shieldu musíme mít shield přesně pro danou desku).

Moduly displejů

Nejběžněji používaným je LCD displej, který je snadno ovladatelný a cenově dostupný. Jedním z hlavních představitelů těchto displejů je jednobarevný znakový 16x2 (16 znaků na řádek a 2 řádky) LCD displej.



Obrázek 8 LCD displej (Arduino LCD displej, 2017)

Tento displej je připojen přes I2C sběrnici. Jedná se o sériovou sběrnici, která připojené zařízení rozděluje do kategorie master nebo slave. Sběrnice je připojena pouze pomocí dvou vodičů. Jeden vodič slouží k přenosu hodinového signálu (SCL – synchronous clock) a je datový kanál (SDA – synchronous data).

2 Realizovaná řešení pro zobrazování dat z Facebook API

Na trhu jsou k dispozici komerční řešení (cena takových zařízení se pohybuje v rozmezí 10-20 000 Kč) nebo vlastní řešení nadšenců, kteří publikují své návrhy a řešení včetně zdrojových kódů a výběru hardwarových prvků.

Flapit

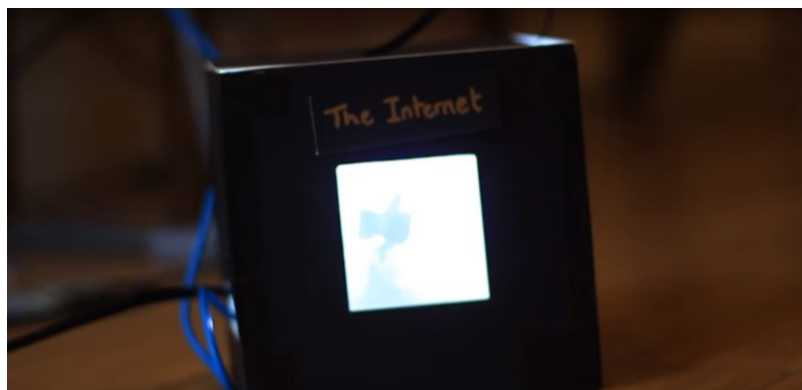
Jedná se o počítadlo statistik sociálních sítí. Je využíváno jako marketingový nástroj k získání nových zákazníků. K internetu se připojuje prostřednictvím Wifi a data zobrazuje na mechanickém číselníku. Toto zařízení je dostupné v obchodním řetězci alza za přibližnou cenou 12 000 Kč.



Obrázek 9 Flapit (Flapit, b.r.)

Arduino Project: Tweet and Facebook Notifier

Jedná se o projekt od pana David J Watts, který navrhl a realizoval zařízení sloužící k upozornění na nový „Like“ na Twitteru nebo Facebooku. Základem toho projektu je Arduino Uno, které je skrze Ethernetový shield připojeno k internetu. Zařízení upozorní na nový „Like“ pomocí pasivního reproduktoru a několika LED, které promítnou siluetu ikony Twitteru nebo Facebooku, jenž jsou zaměřované prostřednictvím servo motoru.



Obrázek 10 Tweet and Facebook Notifier (Arduino Project: Tweet and Facebook Notifier, b.r.)

3 Návrh vlastního hardwarového zařízení

Návrh vlastního zařízení navazuje na práci, která vznikla na Katedře ATŘ (Czudek, 2016). Mé řešení se liší v jiném způsobu zapojení, kdy místo LED driveru, který využil Ing. Czudek použiji 8-bit posuvný registr, jakožto i jinou napájecí část celé sestavy.

Výběr zobrazovací jednotky

Nejprve jsem uvažoval o volbě znakového LCD, u kterých by bylo připojení pomocí I2C sběrnice, tudíž jen za pomoci 2 napájecích a 2 datových žil. Bohužel jsem na českém trhu nenalezl žádný, který by splnil požadavky na co největší možné rozměry z důvodu zobrazování dat ve veřejných prostorech.

Další možností byl grafický LCD MG24065G-SBLWU, s rozměry 180x65x10mm. To by pro zobrazování ve veřejných prostorech mohlo být dostačující, ovšem tento displej by pro připojení k Arduino využil 20 pinů a jeho přibližná pořizovací cena se na českém trhu pohybuje okolo 2 300 Kč. Tudíž ani tento displej není nejlepší variantou.



Obrázek 11 Grafický LCD (MG24065G-SBLWU, 2018)

To mě odradilo od pátrání po dalším LCD a zkusil jsem se soustředit na jinou možnost zobrazovací jednotky. Vzhledem k tomu, že bude potřeba zobrazovat pouze čísla, rozhodl jsem se poohlédnout po 7-segmentovém displeji. Displej jsem hledal s velikostí přibližně 50 mm.



Obrázek 12 7-segmentový displej (LED DISPLAY 56,9MM GREEN HD-AD12RD, 2018)

Nejbližší nalezený rozměr je 57mm, jenž je optimální. V nabídce jsou v červené a zelené barevné variantě. Jelikož půjde pouze o informační zobrazení, nebudu volit červenou barvu, která je většinou využívána pro zobrazení varování. Tudíž použiji 7-segmentový displej se zelenou barvou s označením HD-AD12RD.

Tento displej má LED zapojeny se společnou anodou, tím pádem budou jednotlivé segmenty spínány pomocí zemního vodiče. Po konzultaci s vedoucím práce jsme došli k závěru, že pro zobrazení počtu „Like“ na FB stránkách školy bude 5 číslic dostačujících. Úbytek napětí v propustném směru je v rozmezí 8,4-10 V, ovšem výstup s Arduina je 5 V. Proto bude potřeba zajistit, aby napětí na displeji bylo vyšší než 8,4 V. Nejjednodušším řešením by bylo zapojení pomocí tranzistorů využitých jako spínače. Vzhledem k tomu, že jeden displej bude potřebovat 7 spínaných segmentů (pominu-li nevyužitou tečku) a displejů je 5, dostanu se na 35 tranzistorů, tím pádem i na 35 potřebných výstupů z Arduina. Proto je toto zapojení nevýhodné. Nejlepším řešením je využít posuvný registr. Při volbě bude zapotřebí zohlednit vysoké napětí a velký proud protékající jednotlivými segmenty.

Výběr posuvného registru

Při průzkumu trhu jsem narazil na 8-bit posuvný registr TPIC6B595. Pracovní napětí je v rozmezí 0.3 až 7 V a napětí výstupu může být až 50 V, toto napětí je pro mé zapojení vyhovující.

Maximální možný proud na výstupech je 150mA. Na jeden posuvný registr připojím jeden 7-segmentový displej. Pro zapojení 5 displejů proto bude potřeba 5 posuvných registrů.

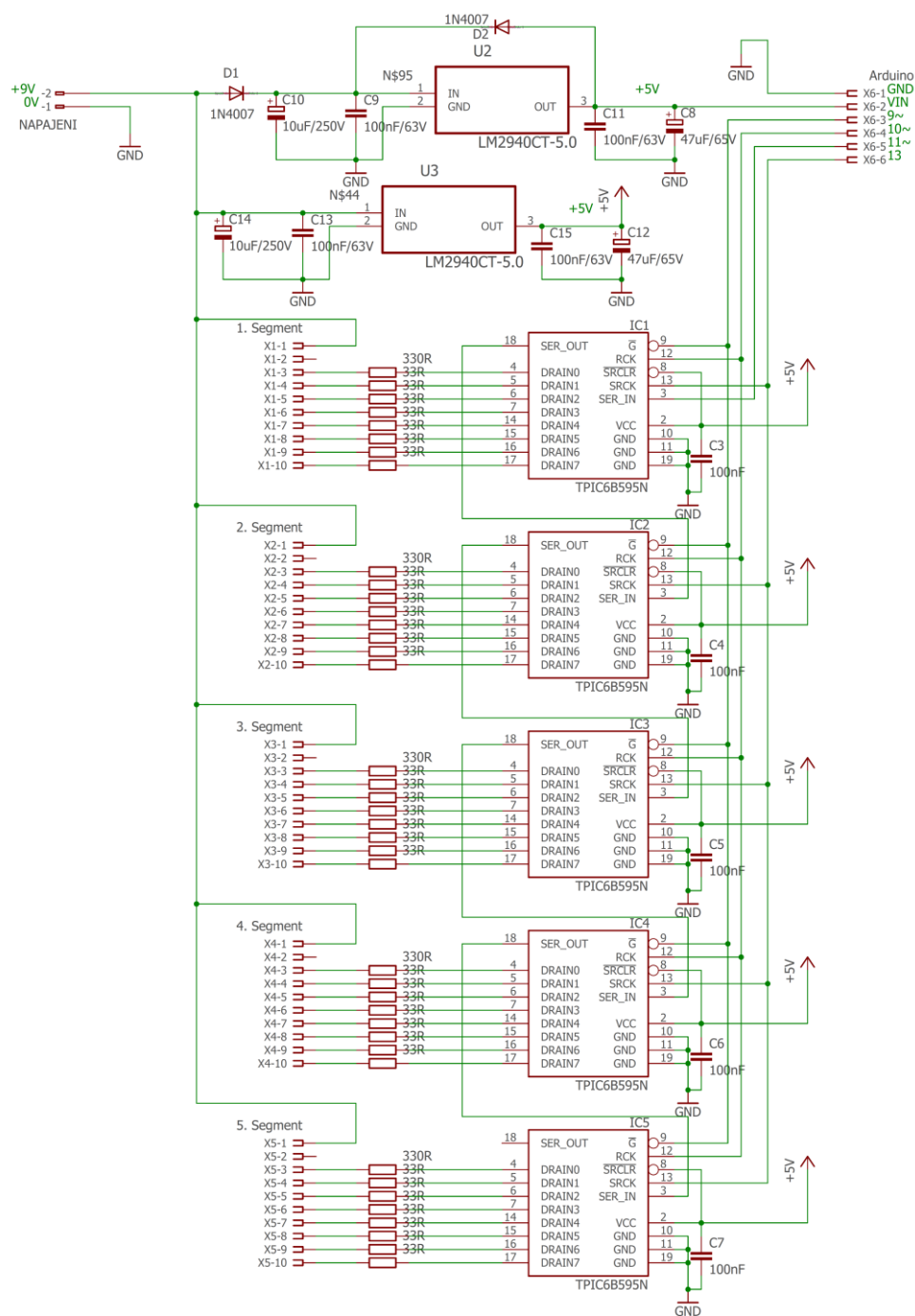
Výběr Arduino desky

Bude potřeba vybrat desku, která bude mít možnost připojení k Wifi. To je možné realizovat pomocí dvou způsobů. Prvním je zvolit Arduino Uno + Wifi shield. Ceny na českém trhu se při této sestavě pohybují přibližně v součtu okolo 3 000 Kč. Druhou variantou je Arduino Yún, které již má vestavěný Wifi modul a jehož cena se pohybuje okolo 2 000 Kč. Vzhledem k ceně a ke kompaktnosti volím Arduino Yún.

Zapojení

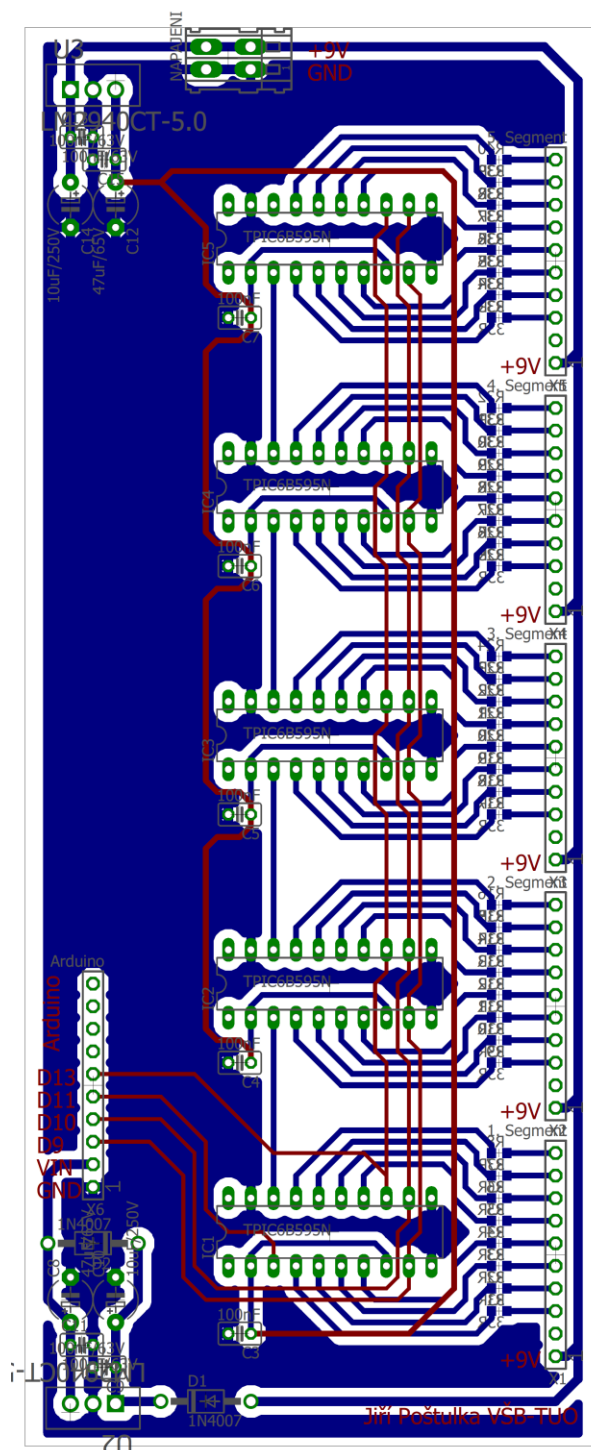
Jak již bylo zmíněno, k ovládání pěti 7-segmentových displejů bude zapotřebí 5 posuvných registrů. Zapojení realizuji dle doporučení datasheetu, hodnotu proudu protékajícího jednotlivými segmenty volím okolo 15 mA, a proto hodnota odporu připojeného na jednotlivé výstupy posuvných registrů bude 330 Ω . Externí napájení bude pomocí síťového adaptéru s výstupem 9V/2A. Bude potřeba toto napětí snížit na 5 V pro napájení posuvného registru a Arduina. Pro regulaci napětí využiji stabilizátor pevného napětí LM2940CT-5, jenž

poskytne výstupní napětí 5V. Tyto regulátory použiji dva, neboť Arduino Yún může mít odběr až 350 mA. Zvolený lineární stabilizátor je až pro odběr 1 A, ale kvůli dlouhodobému provozu použiji zvlášť regulátor pro posuvný registr a pro Arduino Yún. Při stabilizaci přidám k stabilizátoru dva filtrační kondenzátory, jeden před a jeden za stabilizátor, a dva blokovací kondenzátory, které se v případě rychlé změny zatěžovacího proudu vybíjí a tím uvolňují energii. Do zapojení přidám usměrňovací diodu, která bude sloužit k ochraně stabilizátoru, kdy by se při snížení napájecího napětí mohl objevit zpětný proud (z Arduina Yún, pokud by bylo připojeno k PC) a stabilizátor zničit.



Obrázek 13 Schéma zapojení

Po navrhnutí schéma byla následným krokem výroba desky plošného spoje. Nejprve jsem si však funkčnost obvodu pro jistotu zapojil na kontaktním nepájivém poli. Poté jsem mohl navrhnout rozvržení desky, snažil jsem se vše umístit na co možná nejmenší prostor. Současně jsem byl i omezen šířkou desky, neboť jsem nechtěl přesáhnout výšku 7-segmentového displeje v případě, že bych desku umístil hned za displej. Výsledný rozměr desky nakonec dosáhl 160x65 mm. Tento rozměr není příliš malý, bohužel jsem však z důvodu křížení některých cest menšího rozměru nedosáhnul.



Obrázek 14 Deska plošného spoje

4 Analýza možností připojení do Wifi sítě VŠB-TUO

Vysoká škola báňská - Technická univerzita Ostrava poskytuje svým zaměstnancům a studentům možnost bezdrátového připojení (Wifi) do své sítě TUONET. Připojení do sítě je možno realizovat prostřednictvím následujících Wifi bezdrátových sítí (Wifi, b.r.)

Wifi síť EDUROAM

Tato Wifi síť umožňuje připojení do sítě pomocí osobního čísla a nastaveného hesla. Zároveň je možné se pomocí těchto přihlašovacích údajů přihlásit do sítě eduroam i na jiných akademických institucích, které jsou zapojeny do mezinárodního projektu eduroam. Tato síť je dostupná v téměř celém areálu univerzity, proto by byla nejideálnější sítí k připojení Arduina Yún. Zabezpečení WPA2 spolu s podnikovým šifrováním AES potřebuje pro připojení HW i SW podporu zařízení. Díky tomu se Arduino Yún nedokáže na danou síť připojit.

Wifi síť TUONET-SIMPLE

Jedná se o síť určenou pro koncová zařízení, která nepodporují žádné autentizační a šifrovací mechanismy. Z toho vyplývá, že pro připojení Arduina je tato síť jako dělaná, ovšem koncové zařízení vyžaduje od uživatele periodickou autentizaci po uplynutí 20 minut. Tím pádem by pro předvedení zařízení při dni otevřených dveří musela být u zařízení obsluha, která by každých 20 minut zařízení opětovně připojila do sítě. Proto ani tato síť není vhodná.

Wifi síť TUONET-QUEST

Síť je určena pro uživatele s vytvořeným účtem, který má omezenou platnost. Přístup do této sítě je určen pro individuální návštěvy univerzity.

Wifi síť TOUNET-LAB

Pro laboratorní a výuková zařízení, která potřebují, přístup k internetu, byla vytvořena Wifi síť tuonet-lab. Tato síť běží momentálně v pilotním režimu a je dostupná na vybraných místech univerzity. Pomocí vytvořených přihlašovacích údajů vázaných k MAC adrese, je zařízení téměř neomezeno k přístupu na internet. Z toho důvodu se jedná o nejvíce vyhovující síť, kterou univerzita provozuje. A proto pro připojení Arduina Yún v internetu využijí této sítě.

Připojení do Wifi sítě TUONET-LAB

Před připojením zařízení do sítě je potřeba zařízení registrovat. Registraci mohou provést pouze uživatelé se zařazením do LDAP skupiny IOTREG. Registrace je prováděna na základě názvu zařízení, MAC adresy a době platnosti registrace.

IoT zařízení

Heslo pro přístup do sítě: **I6TW-UeBf**

Název	MAC adresa	Platný od	Platný do	IPv4	IPv6
ArduinoYun352	90:a2:da:f7:08:5a	22.11.2017	30.9.2018		

Přidat nové zařízení

Název

MAC adresa

Platný od

Platný do

Hromadné přidání zařízení

Vkládejte data ve formátu "<název zařízení>","<mac adresa (aa:bb:cc:dd:ee:ff)>","<platny-od (dd-MM-yyyy)>","<platny-do (dd-MM-yyyy)>"

CSV soubor Soubor nevybrán

Obrázek 15 Registrace do sítě TUONET-LAB

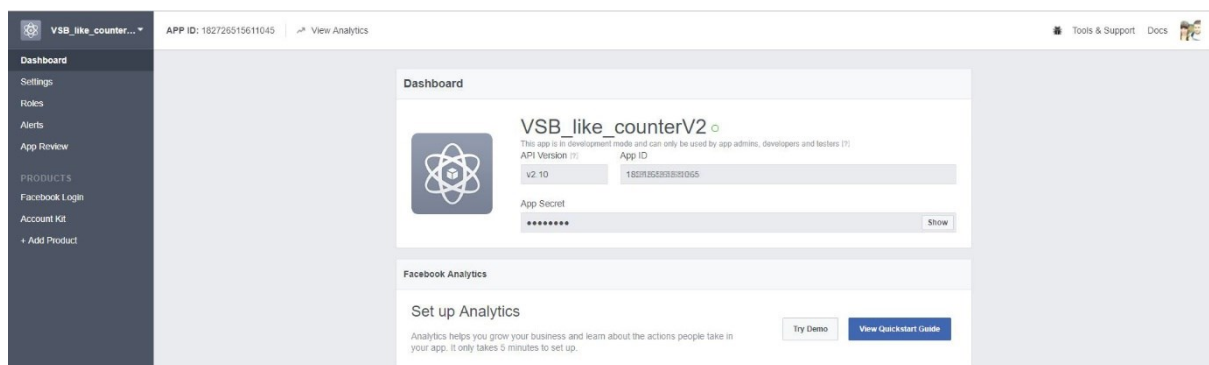
Jak jsem již zmínil, při registraci se uvádí platnost přihlašovacích údajů, která nesmí být delší než 2 roky. Zároveň není možné registrovat již jednou registrované zařízení se stejnou MAC adresou v období, kdy je původní registrace stále v době platnosti. Provoz této sítě je pilotní, tudíž je síť určena pouze pro výuku. Síť není určena pro provoz koncových zařízení, jako jsou mobily, tablety atd. Rychlost připojení je rovněž omezena na 1 Mb/s pro každé zařízení. Na rozdíl od ostatních Wifi sítí VŠB-TUO je umožněna vzájemná komunikace Wifi zařízení.

5 Návrh vlastního softwaru

K realizaci softwaru jsem přistoupil ze směru, který se mi zdál nejlogičtější. Nejprve tedy bude zapotřebí zajistit způsob, jakým získám data z Facebook API, které následovně zpracuji, aby je bylo možné načíst v Arduino. Poté bude nutné tyto data zobrazit na displeji.

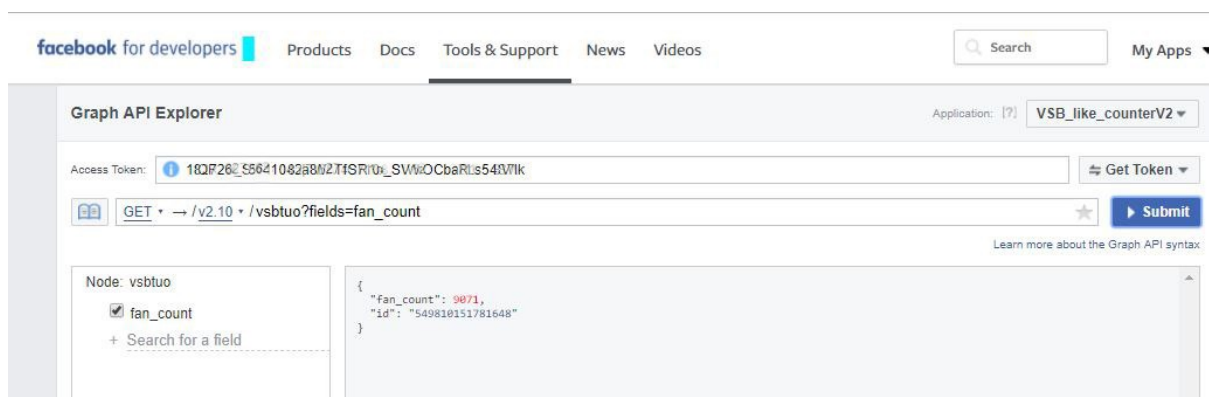
5.1 Facebook API

K přístupu a získávání dat je třeba svůj FB účet převést na Developers, aby bylo možné využít FB API. Pro čtení rozmanitých dat existuje Graph API Explorer, který pomocí Access Tokenu ověří, zda máte k požadovaným datům oprávnění a poté je vygeneruje. Pro získání dat je třeba uživatelský access token nebo app token. Pro mou realizaci jsem zvolil app token, k jehož získání je nutné si založit pod vaším účet aplikaci, která bude tento token obsahovat.



Obrázek 16 Facebook Application

Po úspěšném vytvoření aplikace jsem již mohl přistoupit k Graph API Explorer, který mi pomocí příkazu `vsbtuo?fields=fan_count` a zvolení App access tokenu zobrazil aktuální počet Like.



Obrázek 17 Graph API Explorer

V této aplikaci je možné si i nechat vygenerovat odkaz, který vám po načtení zobrazí právě požadovaný počet Like.

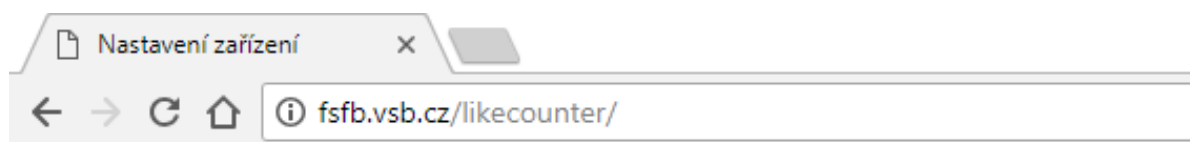


Obrázek 18 Facebookový odkaz

Bohužel tento odkaz obsahuje i zakódovaný access token (z důvodu bezpečnosti mých osobních údajů jsou všechny access tokeny nečitelné). Při pokusu načíst tuto stránku přímo skrze Arduino jej nedokázalo zobrazit právě díky tomuto access tokenu.

5.2 PHP skript

Vyřešení tohoto problému jsem realizoval pomocí dvou PHP skript. Z důvodu následné snazší modifikace jsem se rozhodl vytvořit webovou stránku, ve které je možné zadat FB stránku, na které chceme zobrazit aktuální počet like. Rovněž jsem zde i pro názornost uvedl pár příkladů. Taktéž je zde možné změnit i access token, aby nebylo toto zařízení vázané pouze na jednoho člověka. Zobrazuje se zde i název nastavené FB stránky a aktuální počet like viz obr.19.



Nastavení

Vlož FB stránku: např.: vsb-tuo, kat352, fsostrava...

Vlož access token:

Aktuální nastavení

FB stránka: fsostrava
Počet like: 3137

Obrázek 19 Modifikace zařízení

```

1 <?php
2 $errors = array(); //Pole pro ukládání chyb
3 if(isset($_POST["Save"]))
4 {
5     if (empty($_POST["page"])) { // Nebyla zadána stránka
6         $errors[] = "Nebyla změněna stránka.";
7     }
8     if (empty($_POST["token"])) { // Nebyl zadán token
9         $errors[] = "Nebyl změněn access token.";
10    }
11
12
13    if (!empty($_POST["page"])) { // Ukládání do .txt
14        $fp = fopen("stranka.txt", "w") or exit("Soubor nelze otevřít!");
15        $savestring = $_POST["page"];
16        fwrite($fp, $savestring);
17        fclose($fp);
18    }
19    if (!empty($_POST["token"])) {
20        $fp1 = fopen("token.txt", "w") or exit("Soubor nelze otevřít!");
21        $savestring1 = $_POST["token"];
22        fwrite($fp1, $savestring1);
23        fclose($fp1);
24    }
25 }
26 ?>
27 <!DOCTYPE html>
28 <html>
29     <head>
30         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
31         <title>Nastavení zařízení</title>
32         <h1>Nastavení</h1>
33     </head>
34     <body>
35         <form name="web_form" id="web_form" method="post" action="index.php">
36             <ol><strong>Vlož FB stránku:</strong></label><input type="text" name="page" id="page" /> např.: vsb-tuo, kat352, fsostrava... <br />
37             <strong>Vlož access token:</strong></label><input type="text" name="token" id="token" /> <br />
38             <input type="submit" name="Save" id="Save" value="Submit" /> </ol>
39         </form>
40         <br /><h2>Aktuální nastavení</h2>
41     <?php
42     $fp = fopen("stranka.txt", "r") or exit("Soubor nelze otevřít!"); //Načítání z .txt
43     $text = fread($fp, 10);
44     $b = $text;
45     echo('<ol>FB stránka: <b>' . $text . '</b><br />');
46     fclose($fp);
47
48     $fp1 = fopen("token.txt", "r") or exit("Soubor nelze otevřít!");
49     $text1 = fread($fp1, 50);
50     $d = $text1;
51     fclose($fp1);
52
53     $a = 'https://graph.facebook.com/v2.12/'; // Rozložené řetězce s odkazem od Graph API explorer
54     $c = '/?fields=fan_count&access_token=';
55
56     $homepage = file_get_contents($a . $b . $c . $d);
57     $result = json_decode($homepage, true);
58     echo 'Počet like: <b>' . $result['fan_count'] . '</b></ol>' ;
59
60     if (!empty($errors)) {
61         echo "<ul>";
62         foreach ($errors as $error) {
63             echo "<h5><li>" . $error . "</li></h5>";
64         }
65         echo "</ul>";
66     }
67 ?>
68 </body>
69 </html>

```

V první části je PHP skript, který ověřuje, zda-li bylo něco vloženo. Pokud ano, zkontroluje, jestli byly vyplněny obě pole. Pokud ano, zapíše výsledek do .txt souboru zvláště pro název FB stránky, a zvláště pro access token. Poté následuje sekvence html, která na webové stránce zobrazuje její hlavičku, nadpis a pole pro zadávání textu. Jako poslední sekvence je opět PHP skript, který si otevře výše zmíněný .txt soubor a zobrazí z něj název aktuálně nastavené FB stránky. Pro ověření, že je vše funkční, se php skript pokusí vygenerovat aktuální počet like FB stránky, která je nastavena. V případě že nebylo vyplněno pole FB stránky nebo access tokenu, upozorní na tento stav uživatele na konci stránky.

Druhý PHP skript slouží pro zjednodušení práce Arduino, neboť v případě připojení by první PHP skript načítal veliké množství znaků. Proto sem vytvořil web stránku, na které je zobrazeno pouze ono tížené číslo počtů like.



Obrázek 20 Spuštěný PHP kód

```

1 <?php
2 $fp = fopen("stranka.txt", "r") or exit("Unable to open file!"); //Načtení dat z .txt
3 $b = fread($fp, 10);
4 fclose($fp);
5
6 $fp1 = fopen("token.txt", "r") or exit("Unable to open file!");
7 $d = fread($fp1, 50);
8 fclose($fp1);
9
10 $a = 'https://graph.facebook.com/v2.12/'; //Skládání řetězce odkazu z Graph API Explorer
11 $c = '?fields=fan_count&access_token=';
12
13 $homepage = file_get_contents($a . $b . $c . $d);
14
15 $result = json_decode($homepage, true); //Dekódování URL
16
17 echo $result['fan_count'];
18 ?>

```

Obrázek 21 PHP kód

Skript si nejprve načte data z .txt souboru a poté poskládá řetězec, jehož výsledkem je URL adresa s aktuálním počtem like. Ty poté dekóduje a zobrazí pouze obsah v poli fan_count.

5.3 Arduino

Program pro řízení Arduino se skládá z několika částí. První z nich je část, ve které se inicializují všechny knihovny, jež jsou v programu použity. Další částí jsou pak definované globální proměnné, po kterých následuje první smyčka, která proběhne pouze jednou po spuštění Arduino. V ní se nastavují porty, rychlost sériové komunikace a další nastavení. Po vykonání této smyčky přejde program k hlavní smyčce, která běží po zbytek času, pokud není programem nucena vykonat něco jiného.

Jak jsem již zmínil, první částí programu jsou všechny použité knihovny. V mém programu využívám knihovnu „Bridge“, která má za úkol spuštění a komunikaci s Linuxovým procesorem, skrze který je využívána Wifi. Další knihovnou, kterou jsem použil, je „HttpClient“. Tato knihovna mi mimo jiné umožňuje načítat obsah z http stránek.

```

//=====Knihovny=====
#include <Bridge.h> // Knihovna pro zapnutí Linux procesoru Arduino Yun
#include <HttpClient.h> // Knihovna pro připojení k internetu a stáhnutí obsahu

```

Obrázek 20 Arduino kód – knihovny

Po této části programu následují globální proměnné. První dvě z proměnných definují bytová pole. Tato pole obsahují bity, které jsou zapsány tak, aby v poli 1 byly bity, které na 7-segmentovém displeji zobrazí číslici 1, zatímco v poli Error jsou stejným způsobem nadefinována písmena pro zobrazení hlášky „boot“ a „Error“ na displeji. Pak následuje čtveřice konstant, které definují piny připojené pro komunikaci s TPIC. Zbývající konstanty budou vysvětleny později.

```
//=====Konstanty=====

const byte Císla[] =
{
    B01111110, // 0
    B00011000, // 1
    B01101101, // 2
    B00111101, // 3
    B00011011, // 4
    B00110111, // 5
    B01101111, // 6
    B00011110, // 7
    B01111111, // 8
    B00111111, // 9
    B00000000 // 10
};
const byte Error[] =
{
    B01100111, // E
    B01000001, // r
    B01110001, // o
    B01110011, // b
    B01100011, // t
    B00000000 //
};

const int _7segCLK = 13; // Připoj na TPIC pin 13: SRCLK (aka Clock)
const int _7segLATCH = 10; // Připoj na TPIC pin 12: RCLK (aka Latch/load/CS/SS...)
const int _7segOE = 9; // Připoj na TPIC pin 9: OE (Output Enable)
const int _7segDOUT = 11; // Připoj na TPIC pin 3: SER (aka MOSI)

char hodnoty[5]; // Řetězec do kterého je ukládán počet like
int lost=0;
int g=0;
```

Obrázek 21 Arduino kód - globální proměnné

Jako další proběhne smyčka, ve které nastavuji piny pro komunikaci s TPIC jako výstupní, tudíž budou data pouze vysílat nikoliv však přijímat. Příkaz „Bridge.begin();“ je využit z přidané knihovny a slouží k zapnutí Linux jádra. Tato sekvence trvá delší dobu, proto je před ní vloženo několik příkazů, které mají za úkol poslat na TPIC data, díky kterým se na displeji zobrazí text „boot“, proto aby bylo jasné, že je zařízení funkční a připravuje se ke startu. Poté nastavuji sériovou komunikaci mezi Arduinem a PC na 9600 baudů. Následující příkaz slouží ke kontrole, že doposud vše proběhlo v pořádku a po sériové lince vyšle zprávu „Startuji“.

Posledním příkazem v této smyčce je „delay“, který je nastaven na 5 sekund. Je tam z toho důvodu, že po zapnutí linuxového jádra je potřebná nějaká prodleva pro připojení k Wifi síti.

```
//=====Nastavení=====

void setup() {

    pinMode(_7segCLK, OUTPUT);
    pinMode(_7segLATCH, OUTPUT);
    pinMode(_7segDOUT, OUTPUT);
    pinMode(_7segOE, OUTPUT);

    digitalWrite(_7segLATCH, LOW);          // Řekne TPIC že posílám data

    shiftOut(_7segDOUT, _7segCLK, LSBFIRST, Error[5]); // data na TCIE
    shiftOut(_7segDOUT, _7segCLK, LSBFIRST, Error[4]); // data na TCIP
    shiftOut(_7segDOUT, _7segCLK, LSBFIRST, Error[2]); // data na TCIP
    shiftOut(_7segDOUT, _7segCLK, LSBFIRST, Error[2]); // data na TCIP
    shiftOut(_7segDOUT, _7segCLK, LSBFIRST, Error[3]); // data na TCIP

    digitalWrite(_7segLATCH, HIGH);        // Řekne TPIC že jsou data poslána

    Bridge.begin();                        // Zapínání Linux procesoru

    Serial.begin(9600);                    // Zapnutí seriového portu pro komunikaci s PC, rychlost 9600bps
    Serial.print("Startuji");
    delay(5000);
}
```

Obrázek 22 Arduino kód - smyčka nastavení

Hlavní smyčka programu nejprve inkrementuje proměnou „lost“. V případě, že cyklus proběhne bez navázání kontaktu s http stránkou, vyšle na sériový monitor chybovou hlášku „Connection lost“. Následně skočí do funkce „Chybova_hlaska“. Ta má za úkol poslat na 7-segmentový displej nápis „Error“, který upozorní na problém v komunikaci s internetem. Poté inicializují HttpClient a připojují se na vytvořený PHP skript. Další cyklus načítá znaky z webové stránky tak dlouho, dokud nepřečte všechny. V tomto cyklu znaky rovněž i vysílá na sériový monitor a ukládá je do globální proměnné „hodnoty“. Proměnná „g“ slouží k určení, který znak bude uložen do daného pole v proměnné „hodnoty“, proto se „g“ v tomto cyklu inkrementuje, aby nedošlo k tomu, že se budou znaky neustále přemazávat v jednom poli. Následující podmínka slouží ke kontrole počtu přijatých znaků. Pokud byly pouze 4, doplní v poli 5 znak, který na 7-segmentovém displeji zobrazí volnou pozici. Poté je zavolána funkce „odeslání“.

```
//=====Hlavní smyčka=====

void loop() {

    lost++;

    if (lost>1)                                // Podmínka, která v případě nepřipojení k internetu vypíše chybovou hlášku
    {
        Serial.print("Connection lost");
        Chybova_hlaska();
    }

    HttpClient client;                        // Inicializace HTTP klient

    client.get("http://fsfb.vsb.cz/likecounter/bakalarka.php");
                                           // Nastaví klienta na zadanou URL adresu

    while (client.available())                // Cyklus ověřující jestli je klient připojen
    {
        char c = client.read();               // Zapiše první bit do proměnné c
        Serial.print(c);                     // Vytiskne proměnnou c do seriového portu
        hodnoty[g]=c-'0';
        lost=0;
        g++;
    }
    if (lost==0){
        if(g==4){hodnoty[4]=10;}             // Pokud jsou čísla 4 doplně 5.
        odeslani();
        g=0;
        Serial.println("");                  // V seriovém portu vytvoří nový řádek
        delay(3000);
    }
}
```

Obrázek 23 Arduino kód - hlavní smyčka

Tato funkce slouží pro odeslání dat na TPIC. Nejprve oznámí TPIC, že bude posílat data tak, aby na pin RCLK pošle logickou nulu. Cyklus for pošle přijatá data od poslední číslice až po první, neboť v případě mého zapojení jsou posuvné registry sériově. Co se pošle, to se jako první se zobrazí na posledním 7-segmentovém displeji. Když je pětice bytu poslána, je na RCLK poslána logická jednička, která signalizuje konec posílání dat. Na stejném principu funguje i následující funkce „Chybova_hlaska“. Ta na displeji zobrazí text „Error“ v případě, že Arduino Yún nenaváže kontakt s internetem. Po skončení se program vrací do hlavní smyčky programu.

```
//=====Funkce pro odeslání hodnot do TPIC=====

void odeslani() {
    digitalWrite(_7segLATCH, LOW);           // Řekne TPIC že posílám data
    for(int i=5; i>=0; i--){
        shiftOut(_7segDOUT, _7segCLK, LSBFIRST, Ciska[hodnoty[i]]); // data na TCIP
    }
    digitalWrite(_7segLATCH, HIGH);          // Řekne TPIC že jsou data poslána
}

void Chybova_hlaska() {
    digitalWrite(_7segLATCH, LOW);           // Řekne TPIC že posílám data

    shiftOut(_7segDOUT, _7segCLK, LSBFIRST, Error[1]); // data na TCIP
    shiftOut(_7segDOUT, _7segCLK, LSBFIRST, Error[2]); // data na TCIP
    shiftOut(_7segDOUT, _7segCLK, LSBFIRST, Error[1]); // data na TCIP
    shiftOut(_7segDOUT, _7segCLK, LSBFIRST, Error[1]); // data na TCIP
    shiftOut(_7segDOUT, _7segCLK, LSBFIRST, Error[0]); // data na TCIP

    digitalWrite(_7segLATCH, HIGH);          // Řekne TPIC že jsou data poslána
}
}
```

Obrázek 24 Arduino kód - Funkce

6 Kompletace a popis činnosti

Nejprve jsem hledal již vyrobenou krabičku a to z plastu. Bohužel však rozměry, které potřebuji, nejsou sériově vyráběny. Z toho důvodu jsem se rozhodl pro výrobu vlastní krabičky. Při volbě materiálu jsem se v konečném verdiktu rozhodoval mezi hliníkem a dřevem, po uvážení, že by hliníková krabička nejspíše zcela plně odstínila příjem Wifi signálu, jsem se rozhodl zvolit dřevo. Po výběru desky patřičných rozměrů jsem si ji nejprve nařezal.



Obrázek 25 Nařezaná deska

Po nařezání jsem pro spojení desek nakonec zvolil způsob, kdy jednotlivé desky slepím dohromady. Proto bylo potřeba ještě maličko upravit již nařezané desky, rovněž jsem do přední desky vyřezal otvor pro usazení 7-segmentových displejů. Poté jsem všechny desky přebrousil, aby do sebe jednotlivé díly zapadaly. Následně jsem na jednotlivé díly nanesl lepidlo a celek dostatečně zatížil do doby, než lepidlo zaschlo. Po vytvrzení jsem se nejprve pustil do vybroušení otvorů tak, aby do nich 7-segmentové displeje zapadly s co možná největší přesností.



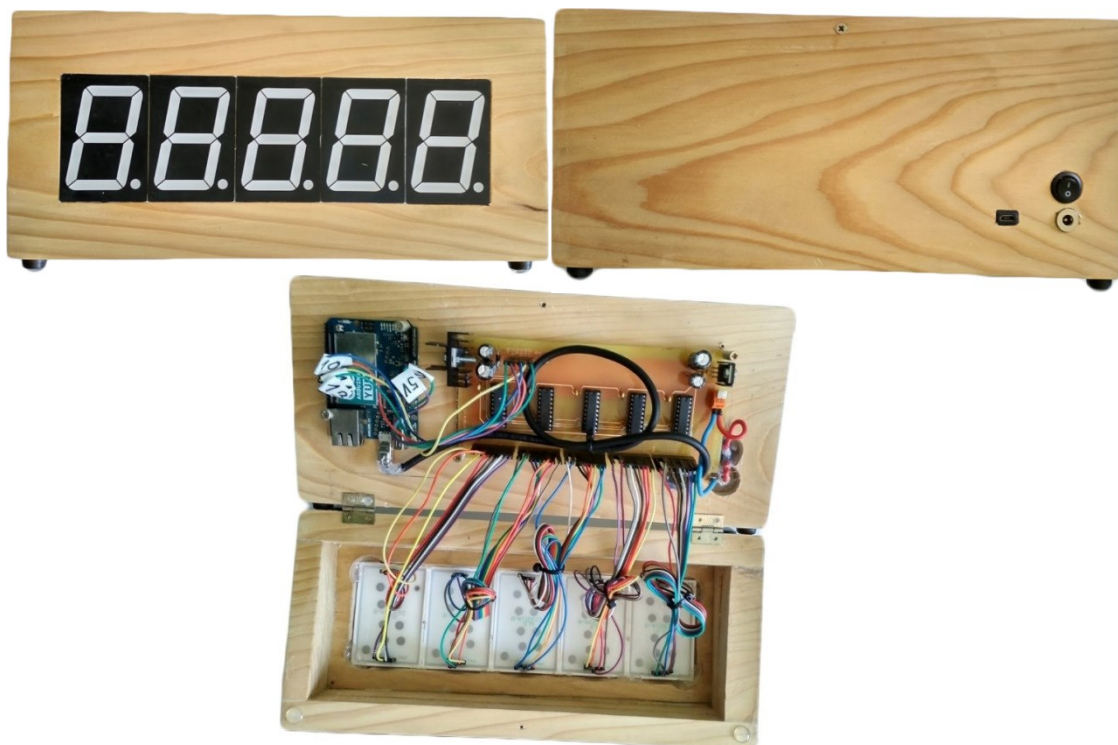
Obrázek 26 Splená krabička

Poté jsem musel přebrousit hrany, které, jak je vidět na předchozí fotografii, přesahovaly. Když jsem měl hotovo, celou krabičku jsem ještě jednou přebrousil několika smirkovými papíry, z nichž tím finálním byl papír o zrnitosti 400, aby byl výsledek co možná nejhladší. Poté, co jsem s tímto skončil, tak jsem ji zbavil nečistot, abych ji mohl natřít bezbarvým lakem.



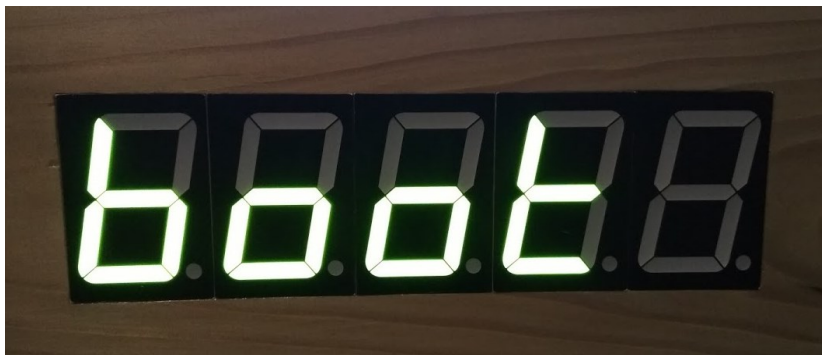
Obrázek 27 Natírání

Po této fázi jsem ji následně nechal dostatečně zaschnout, následně jsem naskládal 7-segmentové displeje do předem připraveného otvoru, abych se ujistil, že pasují přesně. Jako mechanismus otevírání jsem zvolil panty a zabezpečil otevírání zadního víka šroubem. Dalším krokem bylo připevnění na zadní část krabičky Arduino Yún a DPS, kterou jsem předtím vyrobil. Poté jsem vybral vhodné umístění napájecí a programového konektoru, zároveň i kolébkového vypínače. Vyvrtal a vybrousil jsem potřebné otvory. Celou krabičku jsem osadil a drátky propojil napájení desky a 7-segmentové displeje pro komunikaci.



Obrázek 28 Finální vzhled

Pro ověření činnosti je nejprve nutné zkontrolovat, jestli je správně nastavená wifi síť. Postup tohoto nastavení se nachází v přílohách. Po připojení napájení z 9V adaptéru a zapnutí zařízení by se měla na pětici 7-segmentových displejů po pár vteřinách zobrazit hláška „boot“, která značí, že zařízení načítá.



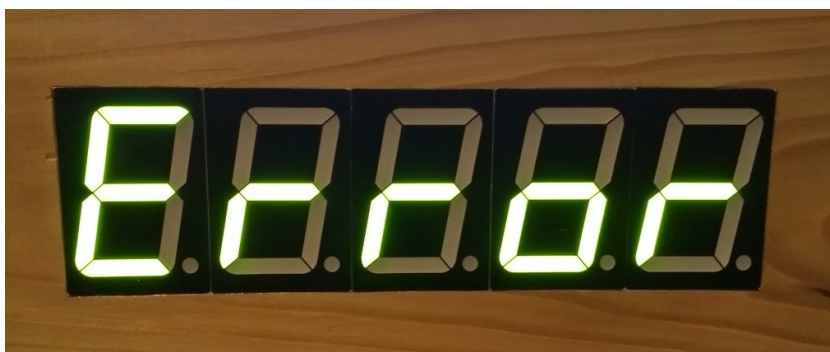
Obrázek 29 Zobrazení "boot"

Poté, co tato sekvence skončí, se na displeji zobrazí aktuální počet like FB stránky univerzity školy.



Obrázek 30 Zobrazení počtu like

V případě, že nebude Arduino schopno připojit se k Wifi, zobrazí na displeji hlášku „Error“.



Obrázek 31 Zobrazení "Error"

7 Závěr

Při vytváření této bakalářské práce jsem se blíže seznámil s vývojovou deskou Arduino, různou škálou typů desek, kterou společnost Arduino nabízí a vývojovým prostředím, dále s možnostmi tyto desky rozšířit o shieldy a moduly.

Následně jsem prozkoumal trh, který nabízí podobné zařízení, jenž je cílem této práce. Poté jsem se nechal inspirovat podobnými projekty využívající desku Arduino a začal navrhovat vlastní realizaci zadané úlohy.

Navrhl jsem schéma, které jsem nejprve ověřil na kontaktním nepájivém poli, poté jsem si nechal na katedře automatizace vyrobit desku plošného spoje, kterou jsem následně vyvrtal a napájel na ni všechny potřebné komponenty.

V tuto chvíli jsem se již mohl pustit do realizace softwaru a jako první jsem hledal způsob, jak data z FB získat. Pro získávání dat existuje webová aplikace Graph API explorer, která mi mou práci výrazně usnadnila. Tato aplikace po zadání přístupového klíče dokáže zobrazit právě i počet Like. A právě na zobrazování tohoto čísla jsem si nechal vygenerovat URL odkaz. Bohužel je však tento odkaz zakódovaný, neboť v sobě obsahuje přístupový klíč. Ten brání, aby se Arduino Yún připojilo přímo na daný odkaz.

Proto jsem si vytvořil dva PHP skripty, které běží na univerzitním serveru. První skript umožňuje nastavení, jaká FB stránka má být zobrazována a rovněž i umožňuje změnu access tokenu, aby nebylo zařízení vázáno pouze na jednoho uživatele. Druhý skript má za úkol načíst a dekodovat obsah z odkazu a poté zobrazit pouze požadované znaky.

Když byl tento problém vyřešen, potřeboval jsem ještě vyřešit, jak toto číslo zobrazit na pěticí 7-segmentových displejů. Proto jsem vytvořil program, který obsahuje dvě smyčky. První z nich načítá data z PHP skriptu, druhá má za úkol tato zpracovaná data poslat na posuvné registry.

Dalším krokem bylo vytvořit krabičku, aby jej bylo možné přenášet a prezentovat na dni otevřených dveří. Jako materiál jsem se rozhodl použít dřevo. Krabičku jsem se snažil vyrobit co možná nejmenší, ovšem limitujícím rozměrem pro mě byla pětice 7-segmentových displejů. Při výrobě jsem rovněž kladl důraz i na vzhled a snadné rozebrání v případě další modifikace.

Další směry využití by mohly vést k větší mobilitě tohoto zařízení pomocí využití napájení z akumulátoru. Obávám se však, že by tento způsob nebyl z dlouhodobého hlediska výhodný, neboť celé zařízení má vysokou spotřebu. Proto bych v tomto případě modifikoval i celé zařízení a použil bych menší displej, rovněž bych se i snažil klást důraz na spotřebu Arduina Yún. Proto bych snížil obnovovací frekvenci a v mezech bych Arduino „uspal“, aby se jeho spotřeba snížila na co nejmenší hodnotu.

Možnou úpravou by samozřejmě byl i displej. Pro navržené schéma by mohly být použity daleko větší 7-segmentové displeje, neboť použité posuvné registry zvládnou daleko větší napětí i protékající proud než ten, který je použit v této sestavě.

Modifikovat by se však nedal pouze HW, ale byla by možná i změna SW. Například by zařízení nemuselo zobrazovat počet like. S drobnou úpravou by mohlo zařízení zobrazovat například teplotu nebo jiné údaje. Zařízení je připraveno zobrazit i tečku, která by byla nezbytná pro zobrazení např. teploty, vlhkosti, znečištění atd.

Domnívám se, že zařízení, které jsem navrhl, je všestranné, co se týče zobrazování veřejných dat z internetu. Díky tomu zařízení nachází mnohonásobně více uplatnění.

Poděkování

Rád bych poděkoval Ing. Pavel Smutný, Ph.D. za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování mé bakalářské práce.

Seznam použité literatury

Arduino - Introduction, 2017. *Arduino* [online]. [cit. 2017-11-25]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction#>

ARDUINO ETHERNET SHIELD, 2017. In: *Arduino* [online]. [cit. 2017-12-07]. Dostupné z: https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/520x330/604a3538c15e081937dbfbd20aa60aad/a/0/a000072_featured.jpg

Arduino LCD displej, 2017. In: *Arduino Project* [online]. [cit. 2017-12-07]. Dostupné z: http://www.arduino-project.cz/520-large_default/arduino-lcd-displej-1602-radic-iici2c.jpg

ARDUINO MEGA 2560 REV3, 2017. In: *Arduino* [online]. [cit. 2017-12-07]. Dostupné z: https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/520x330/604a3538c15e081937dbfbd20aa60aad/A/0/A000067_featured_2.jpg

ARDUINO NANO, 2017. In: *Arduino* [online]. [cit. 2017-12-07]. Dostupné z: https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/520x330/604a3538c15e081937dbfbd20aa60aad/A/0/A000005_featured_2.jpg

Arduino Project: Tweet and Facebook Notifier, b.r. In: *David J Watts* [online]. [cit. 2017-12-22]. Dostupné z: <http://davidjwatts.com/arduino-project-tweet-and-facebook-notifier/>

ARDUINO UNO REV3, 2017. In: *Arduino* [online]. [cit. 2017-11-25]. Dostupné z: https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/520x330/604a3538c15e081937dbfbd20aa60aad/A/0/A000066_featured_2.jpg

Arduino Wi-fi shield, 2017. In: *Arduino.cz* [online]. [cit. 2017-12-07]. Dostupné z: <https://1588504457.rsc.cdn77.org/wp-content/uploads/arduino-wifi-shield-e1440263433797-900x470.jpg>

ARDUINO YÚN, 2017. In: *Arduino* [online]. [cit. 2017-12-07]. Dostupné z: https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/520x330/604a3538c15e081937dbfbd20aa60aad/A/0/A000008_featured_2.jpg

Flapit, b.r. In: *Flapit* [online]. [cit. 2017-12-22]. Dostupné z: https://www.flapit.com/img/flapit_2_alt.png

MG24065G-SBLWU, 2018. In: *GME* [online]. [cit. 2018-01-01]. Dostupné z: https://www.gme.cz/data/product/480_480/pctdetail.513-147.1.jpg

Open-source hardware, 2001. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2017-11-25]. Dostupné z: https://cs.wikipedia.org/wiki/Free_a_open-source_software

SLINTÁK, Vlastimil, 2011. Vývojové prostředí a programování Arduina. In: *UART: Arduino Základy* [online]. Slinták [cit. 2017-11-25]. Dostupné z: <https://uart.cz/90/ide-a-programovani-arduina/>

Typy Arduino desek, 2015. In: *Arduino/Raspberry PI blog* [online]. berkas1 [cit. 2017-11-25]. Dostupné z: <http://www.arpiblog.cz/clanky/typy-arduino-desek>

Seznam příloh

Příloha A - Uživatelský návod

Příloha B - PHP skript 1

Příloha C - PHP skript 2

Příloha D - Arduino kód

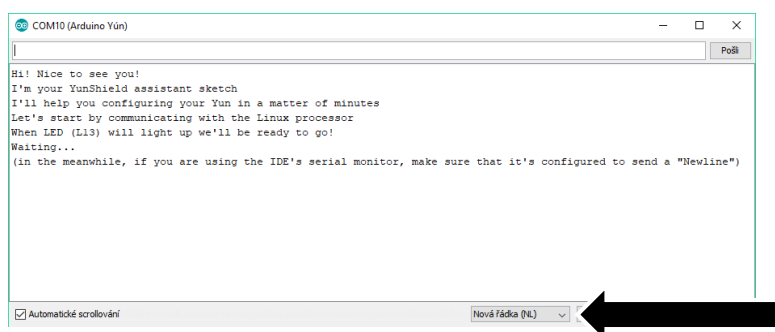
Příloha A – Uživatelský návod

1. Krok – Nastavení Wifi sítě

Wifi síť můžeme nastavit dvěma způsoby:

A) První způsob

Arduino připojíme na PC pomocí USB konektoru a nastavíme správnou desku a port v Arduino IDE. Poté do Arduina nahrajeme příklad, který nalezneme v hlavním panelu pod záložkou Soubor -> Příklady -> Bridge -> YunFirstConfig. Po nahrání tohoto kódu a otevření seriálového monitoru nejprve přepneme do režimu nový řádek, viz obr. 29 a poté budeme požádání k vyplnění údajů, mezi které patří i nastavení Wifi.

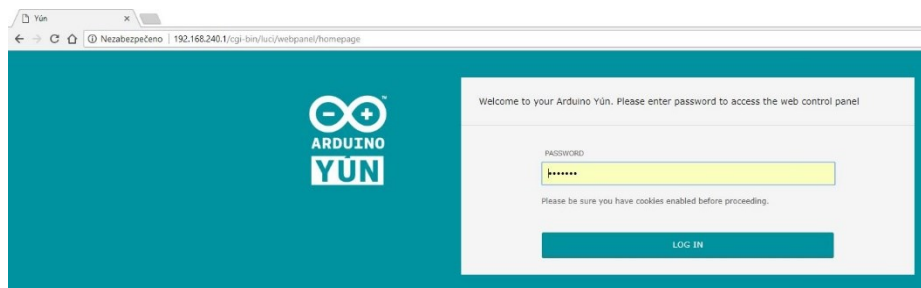


Obrázek 32 Sériový monitor

B) Druhý způsob

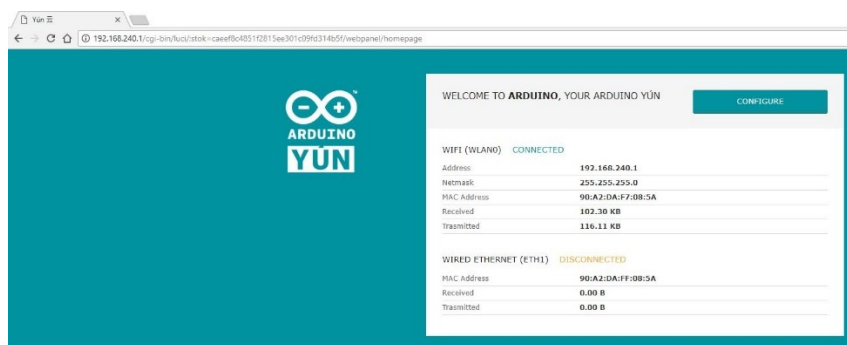
Arduino připojíme k napájení (není nutné, aby bylo připojeno skrze USB k PC). Pokud není Arduino připojeno k wifi síti, mělo by samo vytvořit dostupnou přístupový bod, na který se připojíme. Poté otevřeme internetový prohlížeč a přejdeme na stránku 192.168.240.1.

Pokud je Arduino připojeno k wifi síti, pak musíme být připojeni ke stejné síti a do internetového prohlížeče zadáme IP adresu Arduina, která mu byla přidělena DHCP serverem. Po načtení se nám zobrazí tato stránka, do které zadáme heslo pro přihlášení k Arduinu (výchozí heslo je „arduino“).



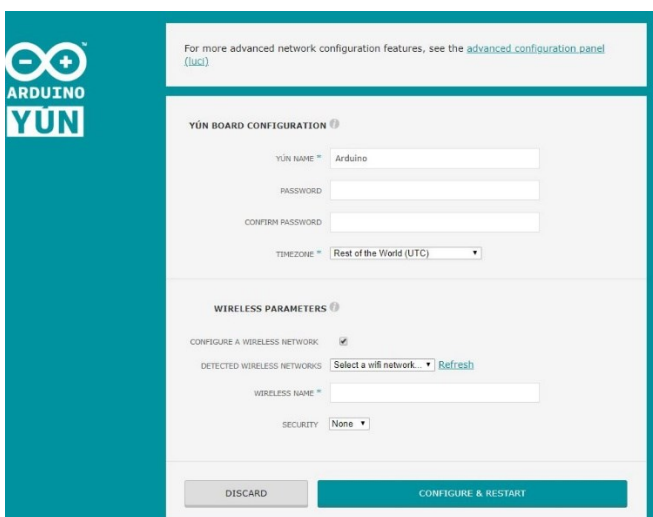
Obrázek 33 Přihlášení do konfigurace

Poté, co se správně přihlásíme, se nám načte stránka, ve které jsou vypsaný všechny aktuálně nastavené wifi sítě.



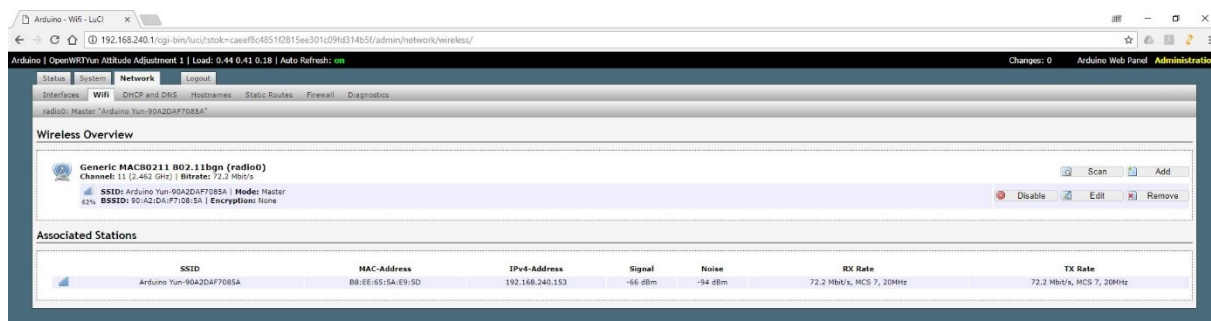
Obrázek 34 úvodní stránka

Následně zvolíme „CONFIGURE“ a v případě, že chceme nastavit pouze jednu wifi síť, zadáme nastavení do této stránky. V případě, že jich chceme nakonfigurovat několik, zvolíme možnost „advanced configuration panel“.

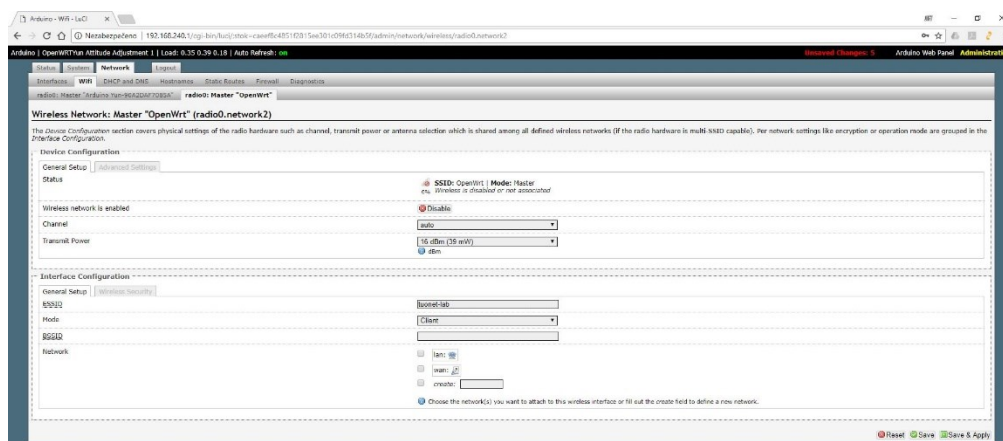


Obrázek 35 První konfigurace

V tomto panelu následně zvolíme záložku Network->wifi a poté dáme „add“

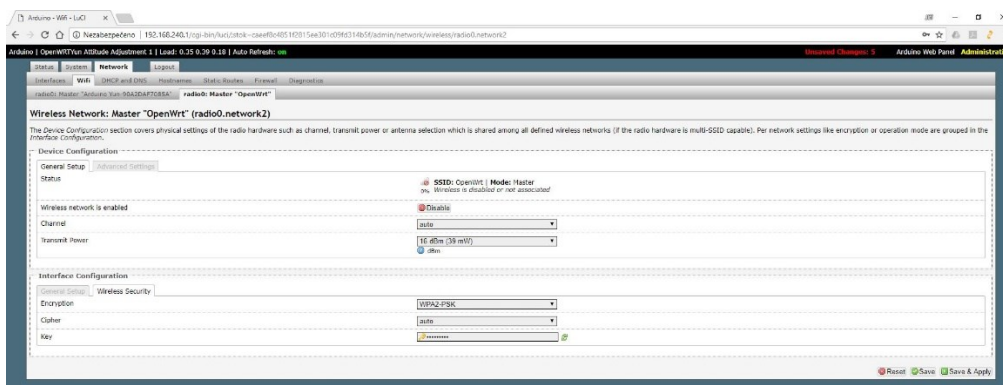


Mode jako client.



Obrázek 36 Nastavení ssid

key napíšeme příslušné heslo.



Obrázek 37 Nastavení hesla

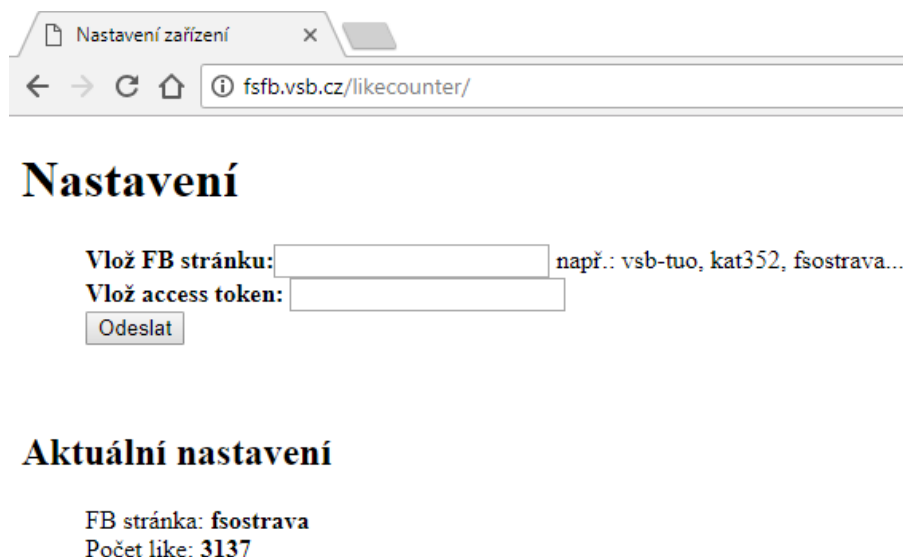
2. Krok – Nahrání arduino kódu

Následující kroky nejsou potřebné, pokud má v sobě Arduino Yún nahraný kód.

V tento okamžik si otevřeme Arduino IDE s kódem, který je vypsán v Příloze D. Připojíme Arduino skrze USB konektor a v Arduino IDE nastavíme jako zařízení Arduino Yún a sériový port, na kterém je Arduino připojeno. Poté dáme tento kód zkompileovat a nahrajeme jej do Arduina. V tento okamžik je zařízení plně funkční a můžeme zařízení odpojit od USB a spustit.

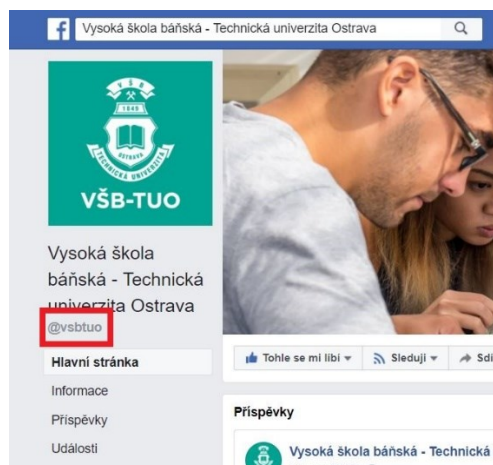
3. Krok – Nastavení parametrů

Po spuštění zařízení je třeba nastavit požadovanou stránku (popř. i vlastní access token), která bude zobrazována na zařízení. Pro snadnou manipulaci s těmito parametry slouží webová stránka, která je dostupná na tomto odkazu <http://fsfb.vsb.cz/likecounter/>.



Obrázek 38 Nastavení parametrů zařízení

Do pole „Vlož FB stránku“ zadáme název FB stránky u které chceme zobrazit počet like na zařízení. Název, který je třeba vložit, nalezneme na titulní straně FB stránky viz Obrázek 39.



Obrázek 39 FB stránka

Do pole „Vlož access token“ zadáme token, který jsme si nechali vygenerovat v podkapitole Facebook API. Následným stiskem „Odeslat“ se data aktualizují a zobrazí se nám aktuálně nastavena stránka a její počet like. Není nutné vyplnit obě pole, v případě že necháme jedno pole prázdné bude brána jeho původní hodnota.

Příloha B – PHP skript 1

```

1 <?php
2 $errors = array(); //Pole pro ukládání chyb
3 if(isset($_POST["Save"]))
4 {
5     if (empty($_POST["page"])) { // Nebyla zadána stránka
6         $errors[] = "Nebyla změněna stránka.";
7     }
8     if (empty($_POST["token"])) { // Nebyl zadán token
9         $errors[] = "Nebyl změněn access token.";
10    }
11
12
13    if (!empty($_POST["page"])) { // Ukládání do .txt
14        $fp = fopen("stranka.txt", "w") or exit("Soubor nelze otevřít!");
15        $savestring = $_POST["page"];
16        fwrite($fp, $savestring);
17        fclose($fp);
18    }
19    if (!empty($_POST["token"])) {
20        $fp1 = fopen("token.txt", "w") or exit("Soubor nelze otevřít!");
21        $savestring1 = $_POST["token"];
22        fwrite($fp1, $savestring1);
23        fclose($fp1);
24    }
25 }
26 ?>
27 <!DOCTYPE html>
28 <html>
29     <head>
30         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
31         <title>Nastavení zařízení</title>
32         <h1>Nastavení</h1>
33     </head>
34     <body>
35         <form name="web_form" id="web_form" method="post" action="index.php">
36             <ol><strong>Vlož FB stránku:</strong></label><input type="text" name="page" id="page" /> např.: vsb-tuo, kat352, fsostrava... <br />
37             <strong>Vlož access token: </strong></label><input type="text" name="token" id="token" /> <br />
38             <input type="submit" name="Save" id="Save" value="Submit" /> </ol>
39         </form>
40         <br /><h2>Aktuální nastavení</h2>
41     <?php
42     $fp = fopen("stranka.txt", "r") or exit("Soubor nelze otevřít!"); //Načítání z .txt
43     $text = fread($fp, 10);
44     $b = $text;
45     echo('<ol>FB stránka: <b>' . $text . '</b><br />');
46     fclose($fp);
47
48     $fp1 = fopen("token.txt", "r") or exit("Soubor nelze otevřít!");
49     $text1 = fread($fp1, 50);
50     $d = $text1;
51     fclose($fp1);
52
53     $a = 'https://graph.facebook.com/v2.12/'; // Rozložení řetězce s odkazem od Graph API explorer
54     $c = '/?fields=fan_count&access_token=';
55
56     $homepage = file_get_contents($a . $b . $c . $d);
57     $result = json_decode($homepage, true);
58     echo 'Počet like: <b>' . $result['fan_count'] . '</b></ol>';
59
60     if (!empty($errors)) {
61         echo "<ul>";
62         foreach ($errors as $error) {
63             echo "<h5><li>" . $error . "</li></h5>";
64         }
65         echo "</ul>";
66     }
67 ?>
68 </body>
69 </html>

```

Příloha C – PHP skript 2

```

1 <?php
2 $fp = fopen("stranka.txt", "r") or exit("Unable to open file!"); //Načtení dat z .txt
3 $b = fread($fp, 10);
4 fclose($fp);
5
6 $fp1 = fopen("token.txt", "r") or exit("Unable to open file!");
7 $d = fread($fp1, 50);
8 fclose($fp1);
9
10 $a = 'https://graph.facebook.com/v2.12/';
11 $c = '/?fields=fan_count&access_token='; //Skládání řetězce odkazu z Graph API Explorer
12
13 $homepage = file_get_contents($a . $b . $c . $d);
14
15 $result = json_decode($homepage, true); //Dekódování URL
16
17 echo $result['fan_count'];
18 ?>

```

Příloha D – Arduino kód

```
//=====Knihovny=====
#include <Bridge.h> // Knihovna pro zapnutí Linux procesoru Arduina Yun
#include <HttpClient.h> // Knihovna pro připojení k internetu a stáhnutí obsahu
//=====Konstanty=====

const byte Cisla[]={
  B01111110, // 0
  B00011000, // 1
  B01101101, // 2
  B00111101, // 3
  B00011011, // 4
  B00110111, // 5
  B01110111, // 6
  B00011110, // 7
  B01111111, // 8
  B00111111, // 9
  B00000000 // 10
};

const byte Error[]={
  B01100111, // E
  B01000001, // r
  B01110001, // o
  B01110011, // b
  B01100011, // t
  B00000000 //
};

const int _7segCLK = 13; // Připoj na TPIC pin 13: SRCLK (aka Clock)
const int _7segLATCH = 10; // Připoj na TPIC pin 12: RCLK (aka Latch/load/CS/SS...)
const int _7segOE = 9; // Připoj na TPIC pin 9: OE (Output Enable)
const int _7segDOUT = 11; // Připoj na TPIC pin 3: SER (aka MOSI)
char hodnoty[5]; // Řetězec do kterého je ukládán počet like
int lost=0;
int g=0;

//=====Nastavení=====
void setup(){
  pinMode(_7segCLK,OUTPUT);
  pinMode(_7segLATCH,OUTPUT);
  pinMode(_7segDOUT, OUTPUT);
  pinMode(_7segOE, OUTPUT);
  digitalWrite(_7segLATCH,LOW); // Řekne TPIC že posílám data
  shiftOut(_7segDOUT,_7segCLK,LSBFIRST,Error[5]); // data na TCIP
  shiftOut(_7segDOUT,_7segCLK,LSBFIRST,Error[4]); // data na TCIP
  shiftOut(_7segDOUT,_7segCLK,LSBFIRST,Error[2]); // data na TCIP
  shiftOut(_7segDOUT,_7segCLK,LSBFIRST,Error[2]); // data na TCIP
  shiftOut(_7segDOUT,_7segCLK,LSBFIRST,Error[3]); // data na TCIP
  digitalWrite(_7segLATCH,HIGH); // Řekne TPIC že jsou data poslána
  Bridge.begin(); // Zapínání Linux procesoru
  Serial.begin(9600); // Zapnutí seriového monitoru pro komunikaci s PC, rychlost 9600bps
  Serial.print("Startuji");
  delay(5000);
}

//=====Hlavní smyčka=====
void loop(){
  lost++;
  if (lost>2){ // Podmínka, která v případě nepřipojení k internetu vypíše chybovou hlášku
    Serial.print("Connection lost");
    Chybova_hlaska();
  }
  HttpClient client; // Inicializace HTTP klient
  client.get("http://fsfb.vsb.cz/likecounter/bakalarka.php"); // Nastaví klienta na zadanou URL adresu
  // Cyklus ověřující jestli je klient připojen
  while (client.available()){
    char c = client.read(); // Zapiše první bit do proměnné c
    Serial.print(c); // Vytiskne proměnnou c do seriového monitoru
    hodnoty[g]=c-'0';
    lost=0;
    g++;
  }
  if (lost==0){
    if (g==2){hodnoty[2]=10;hodnoty[3]=10;hodnoty[4]=10;}
    if (g==3){hodnoty[3]=10;hodnoty[4]=10;}
    if (g==4){hodnoty[4]=10;} // Pokud chybí nějaká čísla tak je doplní prázdným místem
    odeslani();
    g=0;
    Serial.println(""); // V seriovém monitoru vytvoří nový řádek
    delay(3000);
  }
}
```

Obrázek 40 Příloha D - první část arduino kódu

```
//=====Funkce pro odeslání hodnot do TPIC=====
void odeslani(){
  digitalWrite(_7segLATCH,LOW);          // Řekne TPIC že posílám data
  for(int i=5;i>=0;i--){
    shiftOut(_7segDOUT,_7segCLK,LSBFIRST,Cisla[hodnoty[i]]);  // data na TCIP
  }
  digitalWrite(_7segLATCH,HIGH);        // Řekne TPIC že jsou data poslána
}

void Chybova_hlaska(){
  digitalWrite(_7segLATCH,LOW);          // Řekne TPIC že posílám data
  shiftOut(_7segDOUT,_7segCLK,LSBFIRST>Error[1]);  // data na TCIP
  shiftOut(_7segDOUT,_7segCLK,LSBFIRST>Error[2]);  // data na TCIP
  shiftOut(_7segDOUT,_7segCLK,LSBFIRST>Error[1]);  // data na TCIP
  shiftOut(_7segDOUT,_7segCLK,LSBFIRST>Error[1]);  // data na TCIP
  shiftOut(_7segDOUT,_7segCLK,LSBFIRST>Error[0]);  // data na TCIP
  digitalWrite(_7segLATCH,HIGH);        // Řekne TPIC že jsou data poslána
}
}
```

Obrázek 41 Příloha D - druhá část arduino kódu